MOBILE HANDSET WITH BROWSER APPLICATION TO BE USED TO RECOGNIZE TEXTUAL PRESENTATION

Cross Reference To Related Application

[0001] This application is a continuation of U.S. Patent Application 09/463,475 entitled "MOBILE HANDSET WITH BROWSER APPLICATION TO BE USED TO RECOGNIZE TEXTUAL PRESENTATION" and filed on July 27, 2000. The disclosure of the above-described filed application is hereby incorporated by reference in its entirety.

Field of the Invention

[0002] This invention relates to data communications, and in particular to the communication of multimedia documents from multimedia servers to multimedia clients.

Background of the Invention

[0003] The Internet currently provides users with a number of different services. One of such services is the WorldWide Web, or "the Web", which has now become relatively well-known and well-used by fixed terminal subscribers.

[0004] To access the Web, a user installs a "browser" application on a terminal, which acts as a client interfacing with a network of distributed Web servers via the Internet.

[0005] In order to access a page on the Web, a user specifies a universal resource locator (URL), and transmits a page request via a data link to the Internet. In the Internet, the request is routed to the appropriate Web server as identified in the URL. In return, the Web server transmits the pages back to the requesting browser. The communications protocol used within the Internet is TC/IP during this dialogue.

[0006] Web pages are currently formatted using the HyperText markup language (HTML). A Web browser, on receipt of a HTML document, interprets the HTML commands in order to format the text correctly. HyperText links may be included in the document which, when presented to the user, may be selected in order to request a further Web page (a HTML link references a further URL with which the browser retrieves the further page).

Web pages currently contain not only text and hyperlinks, but also image, audio and/or video files (hence, Web pages are referred to as "multimedia" documents).

[0007] Although the textual data in the Web page is generally relatively data-compact, it is recognized that graphics files, audio files, and in particular video files, contain relatively large amounts of data which can reduce related download of Web pages considerably. As a result, the downloading of "true" multimedia documents including graphics, audio and/or video files can be cumbersome, in particular where the data link used into the Internet is of a type which is of a relatively low data rate.

[0008] The speed of download of multimedia Web pages is particularly problematic in the case of mobile data links such as a cellular radio data link, in which the radio data transfer is constrained by the relatively low bandwidth of the radio interface link.

[0009] Therefore, although it is currently possible to access the Web at a mobile communications terminal, the option is currently not widely taken advantage of and, when taking advantage of, the data transferred is generally limited to textual data to achieve reasonable download speeds.

[0010] Methods are known whereby the amount of data transfer involved in accessing the Web are reduced. In particular, many browsers support a caching facility whereby a Web page, once downloaded, is temporarily stored locally on the client terminal. When the Web page is next requested, the browser recognizes, by the URL in the request corresponding with the URL of the cached document, that the document is currently held locally, and retrieves the locally-held page in preference to re-downloading the page from the Web. The browser may transmit an update check to the Web server in question in order to confirm that the locally-stored Web page remains valid, and if not, downloads the updated Web page.

[0011] Caching may be performed on a session-by-session basis, or may be performed such that a cached Web page remains stored between sessions. However, a browser maintains only a limited non-volatile cache of Web pages, and newly-cached Web pages are stored in preference to previously-stored Web pages (which are then deleted) when the cache becomes full.

Summary of the Invention

[0012] In accordance with one aspect of the present invention there is provided a multimedia client terminal, said terminal comprising:

a browser for interpreting a multimedia document received from a remote server, said interpreting means comprising:

means for recognizing textual presentation markup tags in said document and presenting text to a user in accordance with said markup tags;

means for recognizing a standard set of document-independent local library file markup tags in said document;

means for storing a set of non-textual local library files corresponding to said local library tags; and

means for presenting the contents of one of said local library files to a user in response to the recognition of one of said local library file tags in said multimedia document.

[0013] Rather than needing to download non-textual files accompanying the text of a document, the client terminal is able to receive one or more of a known set of document-independent and data-efficient markup tags and access locally-stored files which correspond to the tags in the downloaded document.

[0014] The arrangement of the present invention is to be contrasted with the arrangement in which the client terminal holds a non-volatile cache (although, such a cache may be used in combination with the present invention), in that caching mechanisms do not utilize document-independent local library markup tags. Instead, a document is cached by selection of a URL, and it is unlikely that, when a new multimedia document is downloaded, any parts of the document will already be held locally.

[0015] The use of a non-volatile cache is non-systematic, and does not allow a server content developer to design multimedia documents specifically for low data-rate communication links, which is possible with the present invention.

[0016] In addition, a non-volatile cache arrangement does not allow a user of a low data-rate receiving terminal to be confident that a new multimedia document can be downloaded at an acceptable speed, which is possible with the present invention.

[0017] By defining a standard set of document-independent local library file tags, the present invention allows a server content developer to include one or more of such tags within a document to be placed on a server in the knowledge that a user, provided with a client application for recognizing the predefined set of local library tags and for retrieving appropriate locally-stored files, will be able to download the document relatively quickly and receive each of the intended multimedia parts as intended.

[0018] The present invention is of particular utility in relation to mobile communications terminals.

[0019] Further aspects of the invention are set out in the appended claims.

[0020] Further features of embodiments of the invention are as follows:

1. The system employs tags to locally stored graphical images, drawing primitives, pre-drawn images, image manipulators, audio and video files.

2. A first set of tags provides basic two dimensional drawing functionality, enabling the creation of a range of simple but effective composite images by means of combinations of drawing primitives.

3. The client functionality supports a standard predefined range of pre-drawn images that can add further graphical content to a page, over and above that achieved using the drawing primitives. This enables an image to be retrieved from the local repository to be displayed at the specified co-ordinates. A scaling factor may be specified to dynamically resize the image and a hyperlink can be optionally provided.

4. The client functionality supports the manipulation of on-screen bitmap images.

5. A second set of tags provides basic audio playback capability. The system employs tags to stored speech files as well as the ability to interpret "talking Web pages" through text to speech translation software.

6. The client functionality supports a standard predefined range of audio files enabling a file to be retrieved from the local repository, and spoken by a speech agent at the client.

7. The speech agent supports default vocal and facial styles in order to allow different genders, accents and languages into the playback. Different facial appearances may

be selected for the agent. The combination of attributes may be used to create agents with different personalities.

8. A third set of tags provides basic video clip playback capability. The system employs tags to stored video files.

9. The client functionality supports a standard predefined range of video files enabling a file to be retrieved from the local repository and replayed at the client.

10. In one embodiment, tags implementing the new functionality are embedded into Web pages, which may also contain HTML, and content handlers filter out the tags implementing the new functionality before resolving them.

11. In another embodiment, references to files implementing the new functionality are embedded into Web pages, which may also contain HTML, as markups (e.g. for graphics resources <imgsrc="images/text.hgml">; for audio resources <spesrc="audio/test.hsml>; and for video resources <vidsrc= "video/test.hvml">).

12. In a further embodiment, pages implementing the new functionality are a separate resource, which may be hyperlinked in the corresponding Web pages containing HTML (e.g. http://www.test.com/text.hgml).

13. The drawing functionality provided by tags to graphics primitives supports Web page navigation by means of hyperlinks.

Brief Description of the Drawings

[0021] Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings, wherein:

[0022] Figure 1 shows a schematic block diagram illustrating a document retrieval system in accordance with an embodiment of the invention;

[0023] Figure 2 illustrates a client/server functionality in accordance with an embodiment of the invention;

[0024] Figure 3 illustrates a predefined set of document-independent graphics tags and a set of corresponding graphics elements;

[0025] Figure 4 illustrates an alternative set of graphics elements to be used with the predefined set of graphics tags illustrated in Figure 3; and

[0026] Figure 5 illustrates a screen display provided in accordance with an embodiment of the invention.

Detailed Description of Certain Embodiments of the Invention

[0027] Referring to Figure 1, a multimedia document retrieval system in accordance with an embodiment of the invention includes a mobile client terminal 2 communicating via a mobile communications network 4, such as a GSM digital mobile communications network, and the Internet 6 with the Web server 7.

[0028] The mobile client terminal 2 includes a user interface 8, a client application 10, a radio communications front end 12 and a radio transceiving antenna 14.

[0029] The user interface 8 includes a keypad, a display screen and a loudspeaker for accepting user input and outputting images and audio signals respectively.

[0030] The client application 10 is a customized Web browser, such as a customized version of the Netscape Explorer (Registered Trademark) browser, and includes standard content handlers such as a text content handler 16 and an image content handler 18.

[0031] In addition, the client application 10 includes, as provided in accordance with the present invention, a graphics tag handler 20, an audio tag handler 22 and a video tag handler 24. These content handlers may be components of the browser 10 produced in a development language such as Java, or JavaBeans (Registered Trademark). Associated with each of the tag handlers are a standard graphics library 26, a standard audio library 28 and a standard video library 30, containing content files which are document-independent. The graphics library consists of graphics files GF 1-X. The audio library consists of a number of audio files AF 1-Y, and the video library consists of a number of standard video files VF 1-Z.

[0032] The radio communications front end 12 includes amplifiers, frequency selective circuits, etc which are used to perform radio communications via the radio transceiving antenna 14.

[0033] The mobile client terminal 2 may be one of various types of terminal, for example a laptop computer provided with radio communications facilities, a personal digital assistant (PDA), or a mobile handset provided with Web browsing capabilities.

[0034] Radio signals transmitted and received by the radio transceiving antenna 14 of the mobile client terminal 2 are received and transmitted by a cell site antenna 32 of the mobile communications network 4, via a radio link 33. Only one cell site antenna is illustrated but it will be appreciated that a large number of such antennas are provided in the mobile communications network 4, and the particular serving cell site antenna will depend on the location of the mobile client terminal 2 at any particular time. The cell-site antenna interfaces, via a base station 34, to a mobile switching exchange 36 of the mobile communications network, which maintains a circuit connection 38 with an Internet gateway 40.

[0035] The Internet gateway in turn interfaces via the Internet 6 to the Web server 7, using a TCP/IP connection 42.

[0036] The server 7 holds a set of Web pages 44, each being referenced by different URLs.

[0037] The contents of one such Web page 46 are illustrated in Figure 1. The Web page 46 contains a textual portion 48, to be handled by the text content handler 16, an image file portion 50, to be handled by the image content handler 18, one or more of a standard predefined set of document-independent graphics tags 52, to be handled by graphics tag handler 20, one or more standard predefined document-independent audio tags 54, to be handled by audio tag handler 22, and one or more of a standard predefined set of document-independent video tags 56. Others of the Web pages 44 also contain text content and the graphics tags, audio tags and video tags used in the illustrated document 46 may be repeated in the remaining Web pages of the set 44 held on the server.

[0038] When the client terminal 2 transmits the appropriate URL request to the server 7, via the various data links indicated, the server 7 transmits the Web page back, in HTML format, to the mobile client terminal 2. At the client terminal 2, the various content handlers 16-24 interpret the HTML to present the text and image content 48, 50 of the Web page as indicated by the HTML command therein, and the tag handlers recognize the particular graphics, audio and/or video tags referenced in the document 46, and present the contents of the corresponding file in the libraries 26, 28, 30 as indicated by commands associated with the respective tags.

[0039] Figure 2 illustrates functionality of a client/server at the content handler level, in accordance with one embodiment of the invention. In this embodiment, three different "markup-languages" are provided for developing Web pages on the server site, and which are interpreted by the tag handlers 20, 22, 24 of the system shown in Figure 1. These languages are referred to herein as the hyper-graphic mark-up language (HGML), the hyper-speech mark-up language (HSML) and hyper-video mark-up language (HVML).

[0040] Designed to extend the power of the Web to mobile communications applications, HGML, HSML and HVML provide for fast communications over narrowband radio channels, such as GSM, while also being functional on higher bandwidth systems.

[0041] Through the use of graphical files, audio clips and/or video clips provided on the mobile client terminal in a HGML/HSML/HVML Toolkit, the need for network programming may be significantly reduced, and it is possible to embed tags referencing advanced graphical, speech, video and animation which is stored on the client side into Web pages.

[0042] The content handler interprets the HGML/HSML/HVML commands into graphics primitives and calls to access stored graphical images and speech/video files. It is multi-threaded to allow multiple paints onto the page at a given time.

[0043] HGML, HSML and/or HVML commands may be embedded in a Web page containing HTML, in which case the client side content handlers process the page and filter out the HGML, HSML and/or HVML commands before resolving them.

[0044] References to HGML, HSML and/or HVML resource files may be embedded into Web pages, containing HTML, as markups (e.g. <imgsre-"images/test.hgml">).

[0045] Alternatively, HGML/HSML/HVML pages may be hyperlinked as separate resources in Web pages containing HTML (e.g. http://www.test.com/test.hgml).

[0046] Figure 2 illustrates the case in which the HGML/HSML/HVML resources are embedded as markups. Figure 2 details how content handlers (as exemplified by a HGML content handler) manage content of variable type from a server to display it on a Web page.

[0047] The HGML content handler interprets the HGML commands into either graphics primitive or calls to access stored graphical images. There are a number of areas of functionality (which may be defined in object classes) on the client side:

- **Content handling:** The upper object acting as a container for the other required functionalities;
- **Connection:** To handle the network connection to the server and the retrieval of HGML resources from the server;
- **Command interpreter/Text processing**: Takes the HGML resource and processes it into the various graphics requirements (i.e. if more than one picture resides in the resource, etc); and
- **Painting:** Takes the commands that make a specific HGML graphic and draw the required graphics to the appropriate place in the browser graphical user interface. Again, this could be primitives or stored images. A Paint object extends a thread class and is managed by the command interpreter, allowing multiple paints to be performed at the same time.

[0048] **HGML**

[0049] HGML provides for:

1. Construction of graphics from a standard set of graphics primitives (circles, lines, etc). These are described below.
2. Recall and placement of a standard set of logo/clipart graphics from a client based store. Examples are illustrated in Figures 3 and 4.
3. Prevention of download of photo-realistic images.
4. Yielded graphics which act as hyperlinks.

[0050] HGML includes various tags or commands, which can be grouped into three functional categories, as shown below.

Drawing Primitives	Images	Image Manipulation
ARC	IMAGE	COPY
ARROW	THEME	FLIP
ELLIPSE		PASTE
LINE		ROTATE

LINETO		
LINEREL		
LINK		
ORIGIN		
POLYGON		
RECTANGLE		
SETSTYLE		
TEXT		

[0051] Drawing Primitives

[0052] The first group of document-independent tags provide HGML with basic drawing functionality, enabling the creation of a range of simple but effective composite images.

[0053] Most commands include optional attributes for (out)line color, style, thickness and, when appropriate, fill color. If any of those that are not explicitly specified, the current default settings (as specified by the SETSTYLE tag) are used.

[0054] **ARC**

Draws an arc of a specified width, height and angle from absolute starting coordinates. It has the following attributes:

origin x, y coordinates of arc (absolute) followed by width, height, start angle and arc angle, line color, line style, and line thickness.

e.g. *<arc coords="20,20,50,30,45,80"color=red>*

[0055] **ARROW**

Draws a line from xl,yl to x2,y2 using absolute screen (window) coordinates, with an arrow head at the x2,y2 position. It has the following attributes:

start and end x,y coordinates (absolute), line color, line style, and line thickness.

e.g. *<arrow coords="10,10,100,100",color=black, psize="2">*

[0056] **ELLIPSE**

Draws an ellipse of a given width and height from the specified coordinates. This tag is also used to draw circles. The following attributes are provided:

center x,y coordinates (absolute) followed by width and height, outline color, fill color, outline style, and outline thickness.

e.g. *<ellipse coords="200,200,100,50",color=black;fill=yellow>*

[0057] LINE

Draws a line from x 1 ,y 1 to x2,y2 using absolute screen (window)

coordinates. The attributes are:

[start and end x,y coordinates (absolute), line color, line style, and line thickness.

e.g. *<line coords="10,10,100,100 ",style= "dotted", psize="2">*

[0058] LINETO

Draws a line from the current graphics cursor position to a new position (using absolute coordinates). The following attributes are defined:

end of line x,y coordinates (absolute), line color, line style, and line thickness.

e.g. *<1ineto coords="200,150", color=green>*

[0059] LINEREL

Draws a line relative to the current graphics cursor position using coordinate offsets. The following are its attributes:

end of line coordinates (relative), line color, line style, and line thickness.

e.g. *<linerelcoords="15,-10", psize="4">*

[0060] LINK

Allows a rectangular portion of the screen to be defined as an hyperlink. The following attributes are given:

rectangular region x l,yl,x2,y2 (absolute), hyperlink reference.

e.g. *<link coords="15,10,50,60", href--"http://www.demo.com/ demo. hgm ">*

[0061] ORIGIN

Sets the position of the "graphics cursor". For use with LINETO and LINEREL tags. New graphics cursor position x,y is attributed.

e.g. *<origin coords="100,130">*

[0062] POLYGON

Draws a polygon using a list of absolute screen coordinates, starting with an origin. Coordinates are listed in the format "xl,yl,x2,y2,x3,y3 ...xn,yn". The line from the last specified coordinates back to the origin of the shape is drawn automatically to complete the polygon. The following attributes are provided:

origin of shape, followed by n other points (all absolute coordinates), outline color, fill color, outline style, and outline thickness.

e.g. *<-six-sided-shape->*

<polygon coords= "100,100,120, 80,140, 80,160,100,140,120, 120,120", color=red, fill=red>

[0063] RECTANGLE

Draws a rectangle of a given width and height from the specified coordinates. This tag is also used to draw squares. The following attributes are defined:

top left x,y coordinates (absolute) followed by width and height dimensions, outline color, fill color, outline style, and outline thickness.

e.g. <rectangle coords="10,10,100, 15"style="dotted">

[0064] SETSTYLE

Enables default settings for outline color, fill color and outlines style and thickness to be specified. These settings are then used unless explicitly overridden by the attributes in other HGML commands. The attributes are:

outline color, fill color, outline style, outline thickness.

e.g. *<setstyle color =red, fill=green>*

[0065] TEXT

Displays a text string, starting from the specified coordinates. It has the following attributes:

text position x,y coordinates (absolute), text string to display, and text color.

e.g. *<text coords="10,50'; text="This is an example of HGML text">*

[0066] STORED IMAGES

A standard range of pre-drawn images are stored on the mobile client terminal, associated with a standard document-independent set of graphics tags as exemplified in Fig. 3. These images may be used to provide further graphical content to a page, in addition to that which may be achieved using the drawing primitives.

[0067] IMAGE

Enables an image to be retrieved from the local repository and displayed at the specified coordinates. The current theme will be used unless another one is explicitly

specified. A scaling factor may be specified to dynamically resize the image and a hyperlink can be optionally provided. The attributes are:

image name, theme style, top left coordinates of image, scaling factor (percent), hyperlink reference.

e.g. *<image name=ARROW1, coords="50,50'; scale=150, href="http://www.demo.com/demo.hgm">*

[0068] THEME

Enables a new current image theme to be selected. The attributes are:

name of image theme,

e.g *<theme=DEFAULT>*

[0069] An example of a set of images which may be associated with an alternative theme ("Theme 2"). As shown, different themes may not have defined images for all of the predefined set of tags - some may not be used in associated with a particular theme. Only a set number of theme styles are provided for.

[0070] At a high level, themes are split into three general categories:

- **Navigation**

[0071] Images of buttons for a range of relatively standard Web page functions. The structure of this category is strictly defined by HGML such that images are referenced by standard names and relate to the same type of image irrespective of the theme being used (i.e. HOME, SEARCH, ARROW etc).

- **Design**

[0072] A range of standard page design constructs in terms of lines, bullets, separators, etc. The structure of this category is strictly defined by HGML such that images are referenced by standard names and relate to the same type of image irrespective of the theme being used (i.e. LINE 1. LINE2, BULLET1 etc).

[0073] The navigation and design categories have a prescribed content (i.e. the first element in the Design category is always a line etc). As such, changing the theme associated with a page will mean that, whilst it looks different, the images displayed will still be meaningful and appropriate (e.g. a line will still appear where intended albeit in a different style).

- **User**

[0074] User-defined images of which the content is specific to the theme. HGML defines standard names (i.e. USER1, USER2 etc), but these do not provide any indication of the type of image being referenced.

[0075] The client side HGML functionality provides at least a default theme. Other themes (defined for example by individual site designers) may be downloaded from a remote server the first time a service that uses them is accessed. Alternatively, a user may have a personal preferred theme which may be provided on the client terminal and used in preference to the default and/or downloaded theme.

[0076] User-defined theme elements are provided to permit a degree of flexibility to Web site developers, whilst still remaining optimization advantages.

[0077] This group allows local storage of other images, outside the standard categories prescribed. These may be used to reference any images that the site/theme designer wishes. It is anticipated that this will include images relating to corporate identity (e.g. logos etc) to further standardize the look and feel of their pages. They could, however, also include additional line and bullet styles etc that could not be encompassed under the standard headers provided.

[0078] The number of user-defined images is limited to a predetermined number (e.g. ten) in order to prevent the image repository from becoming undesirably large.

[0079] All images not included in the standard theme may need to be downloaded each time a site is accessed (although they may be cached from a previous session). At such, it is intended that user-defined slots are utilized to hold the largest and most used images (thus helping to further optimize download times).

[0080] Whilst user-defined images are envisaged primarily as site-specific, however other designers may wish to reuse them in their own sites. In this case, there are two alternatives:

(1) to incorporate the required images into the user-defined slots of the site's own theme; or

(2) to make the site reliant on two or more themes (i.e. a default, along with the ones) containing the images to be reused). In this case, the end-user obtains all of the

necessary themes before being able to access the site. The site provider would provide download access for each theme.

[0081] Image Manipulation

[0082] HGML supports the manipulation of the onscreen bitmap image using COPY, PASTE, FLIP and ROTATE commands.

[0083] These are intended to further optimize the creation of certain types of image. For example, if a composite image created using several drawing primitives is also required at another onscreen location, then it may be copied and pasted rather than redrawn. The orientation of it may then be changed using the FLIP and ROTATE operations.

[0084] The functions also help to optimize the storage of theme images. For example, rather than store four Arrow symbols (pointing left, right, up and down), a single image may be stored in the database and then ROTATEd and FLIPed as desired.

[0085] **COPY**

Copies of specified rectangular screen segment to a memory buffer. The original image remains unchanged. The attributes are:

top left and bottom right coordinates of segment.

e.g. *<copy coords="50,50,100,100">*

[0086] **FLIP**

Performs a horizontal or vertical flip of a screen defined by the specified coordinates. The attributes are:

top left and bottom right coordinates of segment. and horizontal or vertical flip,

e.g. *<flip coords="50,50,100,100", axis="horiz">*

[0087] **PASTE**

Copies of the contents of a memory buffer to an appropriately sized rectangular screen segment, starting at the specified coordinates. The attributes are:

top left coordinates to paste segment.

e.g. *<paste coords="100,100">*

[0088] **ROTATE**

Performs a rotation of a circular screen area by a specified angle. The attributes are:

coordinates of center, radius and rotation angle.

e.g. *<rotate coords="50,50,30",angle="80">*

[0089] HSML

Figure 5 illustrates a screen display on the mobile client terminal 2 when displaying a Web page containing HSML.

[0090] As illustrated, the browser displays the text content, formatted as specified by the HTML markups in the Web page, and a speech agent image portion 102, which is driven in accordance with the HSML markups contained or referenced in the downloaded Web page in association with the replay of locally-stored audio clips referenced by audio tags contained in the Web page, or the generation of speech by conversion from text contained in the Web page.

[0091] The speech agent interface provided by the mobile client terminal is controlled by a speech playback/generator program application and a moving image application program stored on the mobile terminal.

[0092] The speech agent has a variety of selectable vocal and facial styles. The various selectable vocal styles are defined in vocal style files stored on the mobile client terminal, containing data describing algorithms and settings for manipulating the audio data providing the speech function to alter the vocal characteristics. The facial style selections are defined in facial style files stored on the mobile client terminal. The facial style files include data defining a plurality of selectable base facial images, a plurality of selectable mouth images, a plurality of image animation algorithms (e.g. talk, smile, wink, etc.) corresponding with the base facial images and the mouth images, and a plurality of image rendering algorithms for rendering the facial style in accordance with selected characteristics of the face, such as skin color, hair color, etc. The vocal and facial style files may be stored permanently on the mobile terminal and/or stored in non-volatile memory on the mobile terminal after being downloaded from an Internet server.

[0093] The speech agent function provided in the mobile client terminal is provided with predefined default settings, in which the vocal and facial style characteristics are preset without input from the user. These characteristics include voice gender, accent, language, base facial style, skin color, hair color, eye color, mouth size, etc. The user may however override the default vocal and facial style characteristics settings, by resetting one or

more of the characteristics via menu options on a man-machine interface of the mobile terminal, in order to provide a personalized default speech agent. The default speech agent settings, whether predefined or user-defined, are used unless an alternative agent theme is specified by mark-ups contained in a Web page, as described below. If an alternative agent theme is specified, the vocal style and the facial style of the speech agent is altered only for the duration of the display of the Web page in question, after which time the default speech agent is reverted to.

[0094] HSML includes the following markups:

Audio Playback	Agent Style	Agent Control
SAY	AGENT THEME	ACTION

[0095] SAY

This markup causes an audio file to be retrieved from, or generated using data retrieved from, the local repository and "spoken" by a speech agent. The default speech agent theme, defining the vocal and facial styles of the speech agent, will be used unless another one is explicitly specified. The attribute required is a phrase name:

[e.g. *<say phrase= "GREETING!">*

"GREETING!" may be a tag to an audio file held in the audio graphics library 28, or may be replayed using a text-to-speech translator on the client side.

[0096] AGENT THEME

This markup causes the vocal and facial style combination for the speech agent to be altered from the default. Voice style attributes may be used in order to incorporate different genders, accents and/or languages into the playback. Face style attributes allow different appearances to be selected for the speech agent. The combination of attributes may be used to create agents with different personalities. The attributes which may be included are:

voice gender, accent, language, base facial style, skin color, hair color, eye color, mouth size, etc.

e.g. *<agent themefemale3,English,Englishfemale5,pale,black,brown, medium>*

[0097] **ACTION**

[0098] This markup controls the image of the speech agent, determining whether it is to be displayed and any movements to be made in conjunction with audio playback. The attributes are an action for the agent (possible actions: appear, talk, smile, wink, etc.) and an action activation status (on/off). e.g. *<action =talk; on>*

[0099] The action markups are referenced by a moving image handler to produce and control the image of the speech agent. When the appear action is activated, the speech agent appears at a specified part of the screen display. When deactivated, the speech agent does not appear. When the talk action is activated, the moving image handler animates the mouth of the speech agent in conjunction with any speech being generated by the SAY command. Other actions (e.g. smile, wink) are also suitably animated when activated.

[0100] **HVML**

[0101] HVML supports a single tag, namely PLAYBACK, which has a file reference attribute. When parsing Web page, the HVML tag is interpreted by the video tag handler, which retrieves a video file from the standard video file library 30 which corresponds with the reference contained in the command. .

[0102] It will be appreciated that the HGML, HSML and HVML functionality illustrated above are provided for exemplary purposes. and are not intended to be limiting. It will be appreciated that various modifications and variations may be employed without departing from the scope of the present invention.